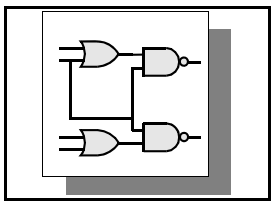
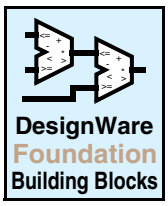




DW01_binenc

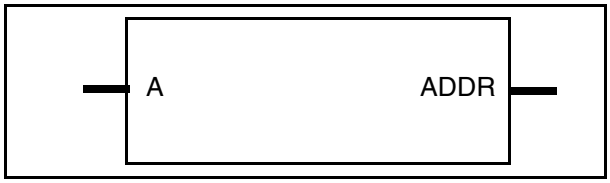
Binary Encoder

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

- Parameterized word length
- Inferable using a function call

Revision History



Description

DW01_binenc encodes the input port A to a binary value on output port ADDR. The encoded value of A is determined by the bit position of the least significant '1' bit. All bits on A higher than the least significant '1' bit are "don't care".

The acceptable values for the *ADDR_width* parameter are greater than or equal to ceil($\log_2$[*A_width*]), as described in Table 1-2. However, the recommended value for *ADDR_width* is ceil($\log_2$[*A_width* + 1]). With this value, the output ADDR can cover all the possible legal values. Note that if *ADDR_width* is equal to ceil($\log_2$[*A_width*]) with *A_width* greater than 1, all 0's on pin A would result in the same ADDR output value as when there is a single '1' bit on A (one-hot encoding) and that '1' is also in the MSB position.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
A	*A_width*	Input	Input data
ADDR	*ADDR_width*	Output	Binary encoded output data

Table 1-2 Parameter Description

Parameter	Values	Description
A_width	≥ 1	Word length of input A
ADDR_width	≥ ceil($\log_2$[*A_width*])	Word length of output ADDR

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Synthesis model	DesignWare
cla[a]	Synthesis model	DesignWare
aot	Synthesis model	DesignWare

a. The ‘cla’ implementation is only available when *A_width* ≤ 512.

Table 1-4 Simulation Models

Model	Function
DW01.DW01_BINENC_CFG_SIM	Design unit name for VHDL simulation
dw/dw01/src/DW01_binenc_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW01_binenc.v	Verilog simulation model source code

Table 1-5 Truth Table (*A_width* = 8, *ADDR_width* = 4)

A(7)	A(6)	A(5)	A(4)	A(3)	A(2)	A(1)	A(0)	ADDR(3:0)
X	X	X	X	X	X	X	1	0000
X	X	X	X	X	X	1	0	0001
X	X	X	X	X	1	0	0	0010
X	X	X	X	1	0	0	0	0011
X	X	X	1	0	0	0	0	0100
X	X	1	0	0	0	0	0	0101
X	1	0	0	0	0	0	0	0110
1	0	0	0	0	0	0	0	0111
0	0	0	0	0	0	0	0	1111

Related Topics

- Logic – Combinational Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Function Inferencing - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use DWARE.DW_foundation_arith.all;

entity DW01_binenc_func is
  generic(func_A_width :integer := 8;func_ADDR_width : integer := 4);
  port(func_A: in std_logic_vector(func_A_width-1 downto 0);
       ADDR_func_TC : out std_logic_vector(func_ADDR_width-1 downto 0);
       ADDR_func_UNS : out std_logic_vector(func_ADDR_width-1 downto 0);
       ADDR_func : out std_logic_vector(func_ADDR_width-1 downto 0));
end DW01_binenc_func;

architecture func of DW01_binenc_func is

begin
  ADDR_func_TC  <= std_logic_vector(DWF_binenc(SIGNED(func_A),
                                    func_ADDR_width));
  ADDR_func_UNS <= std_logic_vector(DWF_binenc(UNSIGNED(func_A),
                                    func_ADDR_width));
  ADDR_func     <= DWF_binenc(func_A,func_ADDR_width);

end func;
```

HDL Usage Through Function Inferencing - Verilog

```
module DW01_binenc_func (func_A,ADDR_func);
  parameter func_A_width = 8;
  parameter func_ADDR_width = 4;

  // Passes the widths to the binary encoder function
  parameter A_width = func_A_width;
  parameter ADDR_width = func_ADDR_width;

  // Please add search_path = search_path + {synopsys_root + "/dw/sim_ver"}
  // to your .synopsys_dc.setup file (for synthesis) and add
  // +incdir+$SYNOPSYS/dw/sim_ver+ to your verilog simulator command line
  // (for simulation).
  `include "DW01_binenc_function.inc"

  input [func_A_width-1:0] func_A;
  output [func_ADDR_width-1:0] ADDR_func;
  assign ADDR_func = DWF_binenc(func_A);
endmodule
```

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_foundation_comp.all;

entity DW01_binenc_inst is
  generic (inst_A_width    : POSITIVE := 8;
           inst_ADDR_width : POSITIVE := 4);
  port (inst_A    : in  std_logic_vector(inst_A_width-1 downto 0);
        ADDR_inst : out std_logic_vector(inst_ADDR_width-1 downto 0));
end DW01_binenc_inst;

architecture inst of DW01_binenc_inst is
begin
  -- Instance of DW01_binenc
  U1 : DW01_binenc
    generic map ( A_width => inst_A_width, ADDR_width => inst_ADDR_width )
    port map ( A => inst_A, ADDR => ADDR_inst );
end inst;

-- pragma translate_off
configuration DW01_binenc_inst_cfg_inst of DW01_binenc_inst is
  for inst
  end for; -- inst
end DW01_binenc_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW01_binenc_inst( inst_A, ADDR_inst );
  parameter A_width = 8;
  parameter ADDR_width = 4;

  input [A_width-1 : 0] inst_A;
  output [ADDR_width-1 : 0] ADDR_inst;

  // Instance of DW01_binenc
  DW01_binenc #(A_width, ADDR_width)
    U1 ( .A(inst_A), .ADDR(ADDR_inst) );

endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
January 2019	DWBB_201806.5	■ Updated example in "HDL Usage Through Component Instantiation - VHDL" on page 5 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com